



REVISTA TRIMESTRAL DA VERYANT E isCOBOL

As rosas são #ff0000, As
violetas são #0000ff, Eu
uso códigos hexadecimais



Mas eu usaria RGB para você.

Obtenha a cor certa em
valores RGB, Hex ou COBOL
padrão usando o utilitário
Color Picker

O isCOBOL tem um utilitário
chamado Color Picker. Você
pode acessá-lo digitando
"CPK" na barra de pesquisa
da área de trabalho do Win-
dows.

Ele o ajudará a escolher a cor
certa todas as vezes e até
mesmo fornecerá o código
necessário para muitos dos
nossos controles de GUI

Confira os detalhes aqui:
[Usando RGB](#)



O inovador da tecnologia COBOL

Seu caminho para o Java

veryant

NOTÍCIAS

NESTA EDIÇÃO

1. O isCOBOL 2025R2 está aqui
2. Programação Moderna com isCOBOL
4. G7 2025 - Você já viu isso? - Foco em TI
5. Adição de Visualizadores de Arquivos ao WebClient
7. Foco no Painel de Pesquisa - Debugging no Thin Client
8. Última página

Grandes notícias da Veryant: O isCOBOL Evolve 2025R2 está aqui!

É com grande satisfação que informamos que a versão mais recente do Evolve — 2025R2 — está oficialmente disponível e repleta de atualizações que tornam o desenvolvimento mais suave, mais rápido e mais moderno.

Aqui está uma visão rápida do que há de novo:

Uma nova perspectiva para o depurador: Adicionamos um tema escuro elegante e um design plano ao depurador gráfico. Isso não é apenas mais fácil para os olhos — também é mais eficiente para os desenvolvedores. **Ferramentas de IDE mais inteligentes:** O IDE agora inclui mais botões da barra de ferramentas para ajudá-lo a alinhar os controles em telas e relatórios rapidamente. Além disso, nossa extensão VS Code oferece suporte a workspace e debugging remoto.

Interface de Usuário (UI) moderna com FlatLaf: Deseja que seus aplicativos tenham uma aparência limpa e contemporânea? O FlatLaf oferece quatro temas elegantes — Light, Dark, Darcula e IntelliJ. E sim, você pode aplicá-los a partir da linha de comando sem alterar o código.

Controles aprimorados: Adicione cores de rollover a visualizações no tree-views, list-boxes e grids. As guias nos controles de abas, agora suportam hints e botões de fechar, e as grids podem ser carregadas dinamicamente para manipular grids conjuntos de dados com mais eficiência. Um novo botão "Today" no date-entry torna a entrada de dados mais rápida e intuitiva.

Aprimoramentos do WebClient: Agora compatível com o Java 21, o WebClient é iniciado mais rapidamente graças ao pré-carregamento de instâncias. Você pode iniciar clusters e sessões com um único comando, e os gestos de pinça facilitam o uso em dispositivos móveis.

Pronto para explorar todos os novos recursos?

Faça o upgrade para o isCOBOL Evolve 2025R2 hoje mesmo e leve seu desenvolvimento COBOL para o próximo nível. Visite o site da Veryant para saber mais e começar a usar.

Modernizando Sintaxes com isCOBOL: Indo Além dos Verbos Cobol

Você acha que o COBOL está congelado nos anos 70, com colunas rígidas e infinitos comando de STRING? Pense novamente.

Atualmente o isCOBOL se desfez de grande parte de sua bagagem legada, e você pode escrever códigos que parecem novos, compactos e até mesmo elegantes. Aqui estão alguns recursos modernos que vale a pena usar em seu próximo programa.

Formato de fonte livre

Esqueça as restrições de colunas. Programe o isCOBOL como qualquer outra linguagem moderna.

Legado:

- Comments start with an asterisk symbol in column 7
- DISPLAY must start at column 12 and end at column 72.
 DISPLAY 'Hello world: This is my first greeting program.'
 'This text goes beyond column 72'.

Moderna:

```
*> Comments can start anywhere using an asterisk and a greater than symbol
*> DISPLAY can start in column 1, 2 or any other column you want
*> and there is no limit at the right side either
DISPLAY 'Hello world: This is my first greeting program. This text goes beyond column 72'.
```

Mais limpo, mais simples, sem colunas para se preocupar.

Diga adeus ao STRING

Por que desperdiçar cinco linhas com uma sintaxe prolixa quando uma basta?

Legado:

```
STRING "Hello, " DELIMITED BY SIZE user-name DELIMITED BY SIZE "!" DELIMITED BY SIZE
      INTO greeting
      END-STRING.
```

Moderna:

```
MOVE "Hello, " & user-name & "!" TO greeting
```

Legível em um piscar de olhos.

Finalmente a memória dinâmica

Pare de codificar limites fixos. Use strings e tabelas flexíveis.

No passado, você tinha que declarar tamanhos muito grandes e fixos — como 900 caracteres ou 900 entradas de tabela — por precaução. Na maioria das vezes, grande parte desse espaço não era utilizado.

Legado:

```
01 fix-string PIC X(900).
01 fix-table PIC 9(3) OCCURS 900 TIMES.
```

Moderno:

```
01 dyn-string PIC X ANY LENGTH.
01 dyn-table PIC 9(3) OCCURS DYNAMIC CAPACITY dt-max.
```

Modernizando Sintaxes com isCOBOL: Indo Além dos Verbos Cobol

Agora o seu código se adapta aos dados reais: dyn-string pode conter 10 caracteres em um caso e 9.000 em outro; dyntable pode armazenar 3 números em determinado momento e 3.000 posteriormente.

Em ambos os casos, a memória é alocada apenas para o que é realmente usado.

Loops que respiram

O loop será finalizado quando a condição for atendida.

Legado:

```
MOVE 0 TO EOF
PERFORM UNTIL EOF = 1
  READ CUSTOMERS NEXT
  AT END MOVE 1 TO EOF
  NOT AT END
  ...
END-READ
END-PERFORM
```

Moderno:

```
PERFORM UNTIL EXIT
  READ CUSTOMERS NEXT
  AT END EXIT PERFORM
  NOT AT END
  ...
END-READ
END-PERFORM
```

Mais legível do que empilhar a lógica em uma única condição.

Funções incorporadas > Chamadas externas

As funções incorporadas podem substituir os CALLs externos. Por exemplo, colocar uma string em caixa alta:

Legado:

```
CALL "TOUPPER"
  USING name, upper-name.
```

Moderno:

```
MOVE $UPPER-CASE(name) TO upper-name.
```

Uma linha simples ao invés de escrever e chamar um subprograma externo.

O que vem a seguir?

O COBOL moderno é conciso, flexível e fácil de desenvolver. E isso é só o começo. Na próxima edição, exploraremos o **isCOBOL orientado para objetos**: sim, classes e métodos, no isCOBOL.

FOCO NA TI

Colocar o foco do programa no painel de pesquisa

Se quiser focar o painel de pesquisa de uma grid, você pode usar o seguinte comando:

```
modify grid1 action  
action-search
```

Este comando é suficiente se o foco estiver na grid. Se o foco estiver em outro controle, será necessário forçar o foco para a grid com as seguintes instruções

```
move grid-id to  
control-id
```

```
move 4 to  
acceptcontrol
```

G7 em Roma!

No mercado globalizado atual, a expansão para diversas regiões com culturas, necessidades e práticas de trabalho diferentes exige mais do que ambição — exige parcerias locais sólidas. Antes de buscar clientes, é essencial identificar parceiros confiáveis que compartilhem suas metas, visão estratégica e conhecimento técnico. Essas alianças abrem portas para mercados que, de outra forma, seriam inacessíveis.

Em 2019, unimos forças com outras empresas para lançar uma iniciativa colaborativa com o objetivo de proporcionar benefícios técnicos e econômicos para todos os envolvidos. Isso levou à criação da "Conferência de Parceiros Estratégicos" anual, organizada rotativamente por cada país participante. O evento concentra-se em tópicos atuais da ciência da computação e oferece uma plataforma para compartilhar experiências e percepções do ano anterior.

Em 2025, a Veryant Italia sediará a quarta edição em Roma, recebendo empresas do Brasil, Filipinas, Japão, EUA, Tailândia, Taiwan e Itália. A conferência, marcada para meados de outubro, destacará dois temas principais: Inteligência Artificial em linguagens de programação e Transformação Digital.



Você viu isso?

Vídeo mais recente:

Fique por dentro dos novos recursos em "Introdução à versão 2025 1".

Saiba como se conectar a fornecedores externos com REST em "Consumindo um serviço REST"

Novos artigos da base de conhecimento:

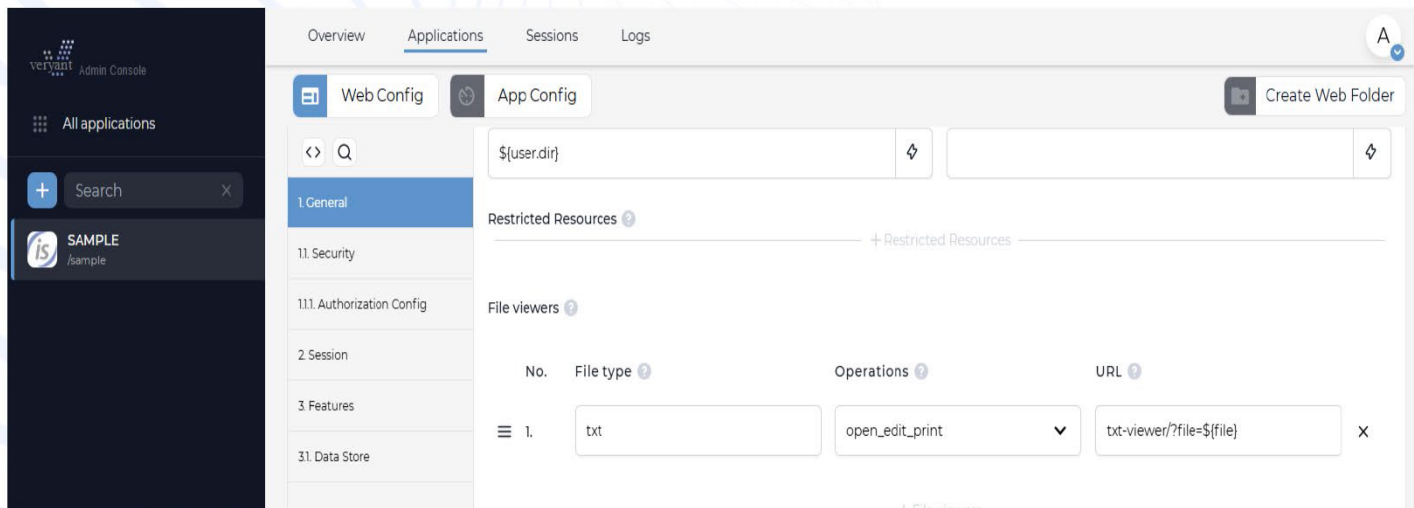
[Como validar um documento XML](#)

[Como alterar a fonte de um programa de caracteres facilmente](#)

Adicionando File Viewers ao WebClient

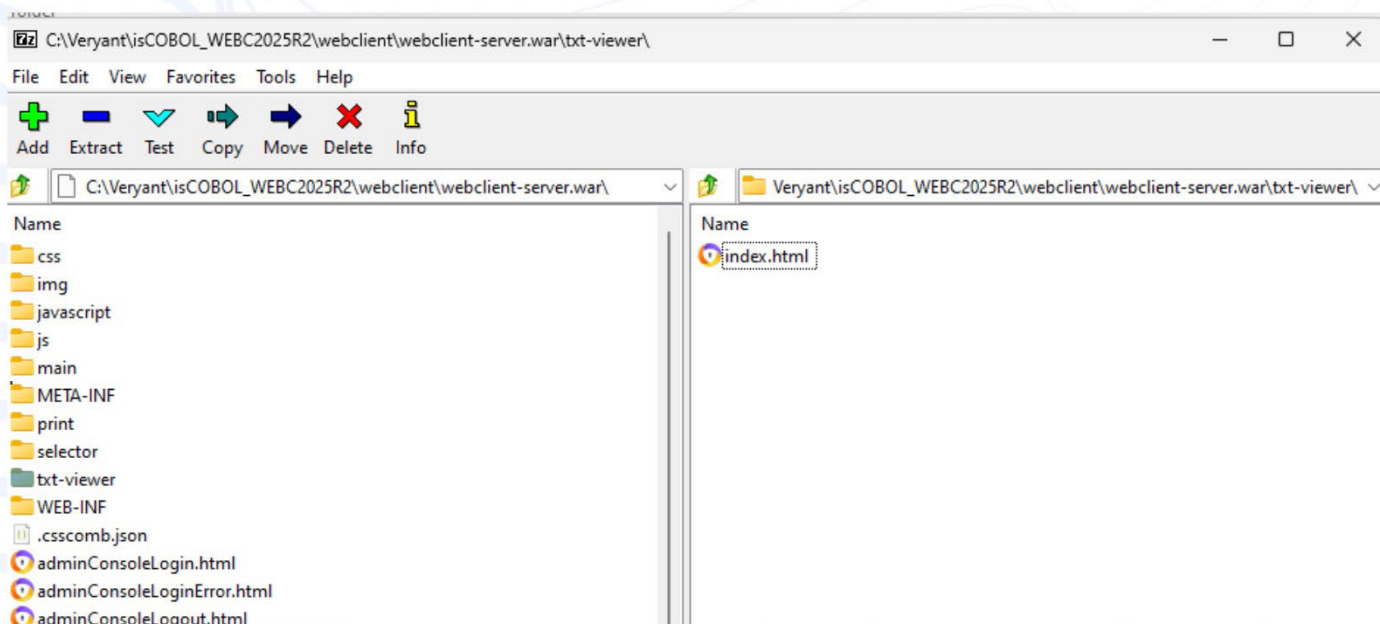
Aplicações COBOL podem gerar arquivos de saída (como .txt ou .pdf) e utilizar a rotina C\$DESKTOP para abrir ou imprimir esses arquivos com suas aplicações associadas.

Na seção “File Viewers” do WebClient, você pode configurar um mapeamento entre extensões de arquivo e URL handlers para gerenciar operações específicas de arquivos do AWT Desktop.



Na configuração mostrada acima, arquivos com extensão .txt são processados usando o /txt-viewer URL handler quando Desktop.getDesktop().open(), edit() ou print() é executado no arquivo.

Para implementar um URL handler, crie uma pasta no diretório webclient-server.war e inclua um arquivo index.html dentro dela.



Adicionando File Viewers ao WebClient

Aqui está um exemplo de um handler de URL para arquivos .txt.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo text file viewer</title>
    <meta charset="utf-8">
    <script language="javascript" type="text/javascript">
      const searchParams = new URLSearchParams(window.location.search);
      const blobId = searchParams.get('file');
      console.log("blobId", blobId)

      document.addEventListener("DOMContentLoaded", function(event) {
        var fileContent = document.getElementById('filecontent');
        fileContent.value = 'Loading file content ...';

        fetch(blobId)
          .then(response => response.blob())
          .then(blob => {
            const reader = new FileReader();
            reader.readAsText(blob);
            reader.onloadend = function() {
              fileContent.value = reader.result;
            };
          })
          .catch(error => {
            fileContent.value = "Blob error " + error
            console.error('blob error', error)
          });
      });
    </script>
  </head>
  <body>
    <h1>Text file viewer</h1>
    <textarea id="filecontent" rows="20" cols="110" style="white-space: pre;font-family:monospace;font-size: 12px" readonly></textarea>
  </body>
</html>
```

Novo e Mais Fácil Sistema de Busca na Documentação Atual do isCOBOL



A partir da versão 2025 R1, a documentação HTML do isCOBOL passou a incluir um recurso de busca aprimorado.

Agora é possível pesquisar substrings utilizando curingas (wildcards), o que é especialmente útil ao procurar propriedades de configuração.

Nas versões anteriores da documentação, a busca exigia informar exatamente o prefixo e conhecer as primeiras letras da propriedade desejada. Por exemplo, para localizar "iscobol.code_prefix", era necessário digitar pelo menos "iscobol.cod" no campo de pesquisa.

Com o novo recurso de curingas, você pode realizar buscas como "code pref"* para encontrar a mesma propriedade. Os resultados são apresentados de acordo com o peso do termo pesquisado. Páginas onde o termo aparece com mais frequência — ou onde ele surge em títulos em vez de apenas em textos de parágrafos — são exibidas primeiro.

Boas práticas para Debugging em Thin Client



Os programas executados em um ambiente de produção devem sempre ser compilados em release-mode, não em debug-mode. Mas e se houver vários usuários ativos conectados ao isCOBOL Server e um deles relatar um comportamento estranho que você queira investigar? Como proceder para debug de um ou mais programas sem afetar os demais usuários?

Nesse cenário, você pode pensar que a melhor opção seria copiar as debug-compiled classes para a pasta de programas do ambiente de produção, sobrescrevendo as release-compiled classes existentes, a fim de executar um cliente em debug mode. Depois, seria necessário restaurar as release-compiled classes quando o seu debugging fosse concluído.

No entanto, essa abordagem possui um efeito colateral muito importante: **classes compiladas em debug-mode não são totalmente thread safe e podem gerar memory leaks** quando são carregadas por clientes que não estão rodando em debug mode e utilizam a JVM principal do isCOBOL Server.

Isso pode causar problemas sérios, especialmente se o isCOBOL Server for reiniciado com pouca frequência.

Uma prática mais adequada para debugging em ambientes de thin client é isolar as classes compiladas em modo debug, seja utilizando um code-prefix separado, seja por meio de um servidor isCOBOL dedicado. Esses procedimentos são explicados detalhadamente na documentação em isCOBOL Application Server: Boas Práticas para o Uso de Programas Compilados em Modo Debug.

Junte-se a nós no

Twitter, LinkedIn ou Facebook para ficar por dentro das novidades da Veryant



Assista aos nossos vídeos de demonstração e inscreva-se no nosso canal do YouTube



Vampiro



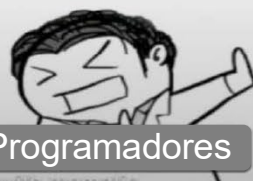
Luz solar



Super-homem



Kryptonita



Programadores

Documentação



Evolução sem revolução



Entre em contato:

Para o e-mail do suporte ao cliente você pode entrar em contato conosco pelo e-mail support@veryant.com

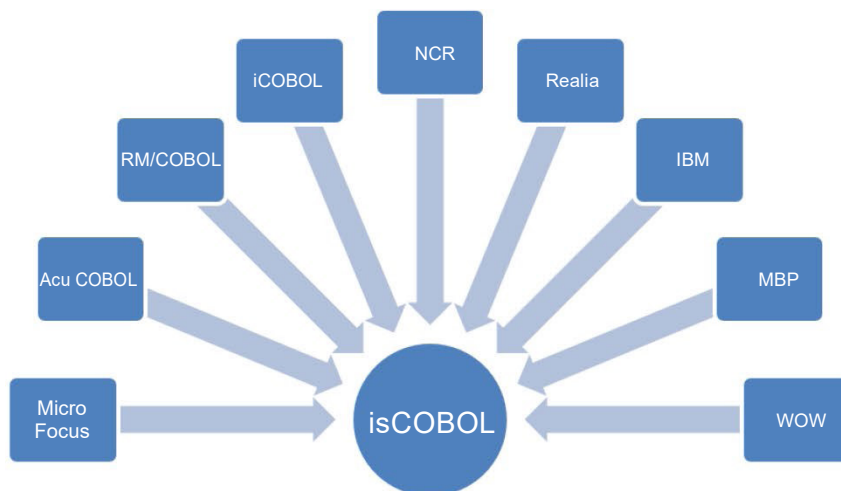
Se quiser que a Veryant entre em contato com você para agendar um briefing técnico sobre o produto, envie um e-mail para info@veryant.com

Se quiser que a Veryant entre em contato com você para obter uma cotação especial ou assistência de vendas, envie-nos um e-mail para sales@veryant.com

Sede corporativa
6390 Greenwich Dr., Suite 225
San Diego, CA 92122 - EUA
Tel. (inglês): +1 619 797 1323
Tel. (Espanhol): +1 619 453 0914

Sede na Europa
Via Pirandello, 29
29121 - Piacenza - Itália
Telefone: +39 0523 490770
Fax: +39 0523 480784
emea@veryant.com

Como sempre, a versão mais recente do isCOBOL Evolve traz diversas melhorias de compatibilidade, reforçando nosso compromisso em tornar o seu processo de conversão cada vez mais simples, rápido e eficiente.



veryant.com

Siga a **Veryant** no



veryant.com

©2025 Veryant - Todos os direitos reservados.